*5λ*

# UNITED STATES PATENT AND TRADEMARK OFFICE

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/837,036 | 04/18/2001 | Tse-Hua Lan | US 010202 | 7522 |

| 24737 | 7590 | 08/19/2004 |
|---|---|---|

PHILIPS INTELLECTUAL PROPERTY & STANDARDS
P.O. BOX 3001
BRIARCLIFF MANOR, NY 10510

| EXAMINER |
|---|
| WONG, ALLEN C |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2613 | |

DATE MAILED: 08/19/2004       *7*

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

| | Application No. | Applicant(s) |
| --- | --- | --- |
| **Office Action Summary** | 09/837,036 | LAN ET AL. |
| | **Examiner** | **Art Unit** |
| | Allen Wong | 2613 |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

## Period for Reply

**A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.**
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

1)☒ Responsive to communication(s) filed on _04 June 2004_.

2a)☒ This action is **FINAL**.     2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

4)☒ Claim(s) _1-5 and 7-30_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-5,7-15 and 22-24_ is/are rejected.

7)☒ Claim(s) _16-21 and 25-30_ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

## Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All  b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

U.S. Patent and Trademark Office

PTOL-326 (Rev. 1-04)      Office Action Summary      Part of Paper No./Mail Date 7

## DETAILED ACTION

### *Response to Arguments*

1.      Applicant's arguments filed 6/4/04 have been fully read and considered but they are not persuasive.

The minor claim objections to claims 4, 13, 14 and 25 have been withdrawn since applicant has properly amended the claims.

Regarding lines 9-13 on page 15 cf applicant's remarks, applicant states that claim 1 has been amended. The examiner acknowledges that claim 1 is no longer anticipated under Florencio, since Florencio does not specifically disclose "a computation load from header information selected from the group consisting of: macroblock-type, motion vector magnitude, motion vector count, non-zero discrete cosine transformer coefficients and a coded block pattern (CBP)".

However, this limitation is well known to one of ordinary skill in the art, but if one is not convinced, then one can peruse Wittig since Wittig teaches the extracted header information includes a macroblock type, a motion vector magnitude, a motion vector count, non-zero discrete cosine transformer(DCT) coefficients, and a coded block pattern (CBP) number from said decoded block-based data packets (col.3, ln.9-14; Wittig discloses the header information contains macroblock type, motion type and a plurality of motion vectors which includes motion vector magnitude and count, DCT type which includes non-discrete cosine transformer coefficients, and a coded block pattern number).

Therefore, it would have been obvious to one of ordinary skill in the art to combine Wittig's teaching of utilizing the header information with Florencio's video decoding method and system for obtaining relevant header information so as to decode MPEG video data streams more efficiently and quickly, and at the same time, preserving high image quality (col.1, ln.44-50).

With regards to claims 2, 3 and 7, they are rejected for the same reasons as independent claim 1.

Regarding lines 23-24 on page 16 of applicant's remarks, applicant contends that there is no motivation found in Florencio and Tan. The examiner respectfully disagrees.

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, it would have been obvious to one of ordinary skill in the art to combine Tan's complexity estimator with Florencio's video decoder method and system for properly decoding as much video data as possible without losing pertinent information, and for versatile decoding in multiple formats and resolutions, as disclosed in Tan's col.8, ln.63 to col.9, ln.4.

Regarding lines 24-26 on page 16 of applicant's remarks, applicant argues that even if the teachings were combined, the combined device would not include, teach or

suggest ail the claim limitations recited in the claims. The examiner respectfully

disagrees. The test for obviousness is not whether the features of a secondary

reference may be bodily incorporated into the structure of the primary reference; nor is it

that the claimed invention must be expressly suggested in any one or all of the

references. Rather, the test is what the combined teachings of the references would

have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208

USPQ 871 (CCPA 1981).

Regarding lines 13-14 on page 17 of applicant's remarks, applicant states that

Tan's decoding unit does not perform any calculation regarding the processing or the

complexity of the image. The examiner respectfully disagrees. In col.8, ln.23-60, Tan's

fig.1, element 180 has the complexity estimator is applied to the decoder 190 and a

feedback loop is used to recursively estimate the complexity or to *selectively adjust* or

*predict* the complexity, ie. the computational load of the IDCT and motion generating

means, of the next decoded picture, wherein fig.6, the specifics of the complexity

estimator are further disclosed in that the error 186 of the complexity measure is used

by calculating the difference between the estimated complexity signal 184 and the

actual complexity signal 185, and that the error 186 goes back to the feedback gain

187, and then the output of the feedback gain eventually goes back to the complexity

estimator 183 for constant recalculation of the complexity or the computational load of

the IDCT and the motion generating means, and the complexity information is

implemented into decoder 190 of fig.2 where the components of the decoder 190 are

affected or *scaled*, so the IDCT 194 is *scaled* and the motion generating means 196 is *scaled*. Thus, the computational load of the MPEG decoder is *selectively adjusted*.

Regarding lines 27-28 on page 17 of applicant's remarks, applicant asserts that neither Florencio nor Tan suggest any operation at the decoder for determining computation loads based on the information in the transmitted header. The examiner respectfully disagrees. In col.8, ln.23-60, Tan's fig.1, element 180 has the complexity estimator is applied to the decoder 190 and a feedback loop is used to recursively estimate the complexity or to *selectively adjust* or *predict* the complexity, ie. the computational load of the IDCT and motion generating means, of the next decoded picture, wherein fig.6, the specifics of the complexity estimator are further disclosed in that the error 186 of the complexity measure is used by calculating the difference between the estimated complexity signal 184 and the actual complexity signal 185, and that the error 186 goes back to the feedback gain 187, and then the output of the feedback gain eventually goes back to the complexity estimator 183 for constant recalculation of the complexity or the computational load of the IDCT and the motion generating means, and the complexity information is implemented into decoder 190 of fig.2 where the components of the decoder 190 are affected or *scaled*, so the IDCT 194 is *scaled* and the motion generating means 196 is *scaled*. Thus, the computational load of the MPEG decoder is *selectively adjusted*.

Regarding lines 29-30 on page 17 of applicant's remarks, applicant contends that there is no motivation found in Florencio and Tan. The examiner respectfully disagrees.

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, it would have been obvious to one of ordinary skill in the art to combine Tan's complexity estimator with Florencio's video decoder method and system for properly decoding as much video data as possible without losing pertinent information, and for versatile decoding in multiple formats and resolutions, as disclosed in Tan's col.8, ln.63 to col.9, ln.4.

Regarding line 30 on page 17 to line 3 on page 18 of applicant's remarks, applicant argues that even if the teachings were combined, the combined device would not include, teach or suggest all the claim limitations recited in the claim 8. The examiner respectfully disagrees. The test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference; nor is it that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981).

With regards to claims 9-13, they are rejected for the same reasons as independent claim 8.

Regarding lines 24-27 on page 18 of applicant's remarks, applicant argues that neither Florencio nor Tan teach a decoding system determining a complexity estimator and a device formed from the combined teachings would not perform a determination of a complexity level at the decoder. The examiner respectfully disagrees. As stated above, in col.8, ln.23-60, Tan's fig.1, element 180 has the complexity estimator is applied to the decoder 190 and a feedback loop is used to recursively estimate the complexity or to *selectively adjust* or *predict* the complexity, ie. the computational load of the IDCT and motion generating means, of the next decoded picture, wherein fig.6, the specifics of the complexity estimator are further disclosed in that the error 186 of the complexity measure is used by calculating the difference between the estimated complexity signal 184 and the actual complexity signal 185, and that the error 186 goes back to the feedback gain 187, and then the output of the feedback gain eventually goes back to the complexity estimator 183 for constant recalculation of the complexity or the computational load of the IDCT and the motion generating means, and the complexity information is implemented into decoder 190 of fig.2 where the components of the decoder 190 are affected or *scaled*, so the IDCT 194 is *scaled* and the motion generating means 196 is *scaled*. Thus, the computational load of the MPEG decoder is *selectively adjusted.*

With regard to claims 15-21, claim 15 depends on independent claim 14, and thus it is rejected for the same reasons as independent claim 14. Claims 16-21 are already objected to as being indicated with allowable subject matter.

Regarding lines 8-10 on page 19 of applicant's remarks, applicant states that

since the remarks for claims 8 and 14 are appropriate, claim 22 is allowable for the

same reasons as claims 8 and 14. The examiner respectfully disagrees. Independent

claim 22 is also rejected for the same reasons as independent claims 8 and 14, as

stated in the above paragraphs.

With regards to claims 23-30, claims 23-24 are rejected for the same reasons as

independent claim 22. Claims 25-30 are already objected to as being indicated with

allowable subject matter.

With regards to claims 4-5, since these claims depend from independent claim 1,

there are rejected by virtue of their dependence from a rejected base claim.

Thus, rejection of claims 1-5,7-15 and 22-24 are maintained. Claims 16-21 and 25-

30 are still objected to as being indicated with allowable subject matter not found in the

prior art.

### Claim Rejections - 35 USC § 103

1.      The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

2.      Claims 1-3 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable

over Florencio (6,125,147) in view of Wittig (6,011,498).

Regarding claim 1, Florencio discloses an MPEG decoder for improving

decoding efficiency by scaling the decoding of an encoded digital video signal (see

fig.2 and col.2, ln.28-29), comprising:

means for decoding a compressed video data stream including a plurality of

macroblocks, said decoding means being operative to output quantized data from said

decoded data stream (col.6, ln.25-27 and fig.2, element 212 is a variable length

decoder, VLD, that decodes compressed data and outputs quantized data);

means for performing an inverse quantization operation on said quantized data

(col.6, ln.28-31 and fig.2, element 213);

means for performing an inverse discrete cosine transform (IDCT) of the output

from said means for performing an inverse quantization operation (fig.2, element 214);

means for extracting header information from said quantized data from said

decoding means and for performing a prediction operation according to predetermined

criteria (col.6, ln.62 to col.7, ln.2-8; the variable length decoder (VLD) 212 in decoder

210 extracts the header information from the quantized data and performs a prediction

operation, by using this header information which contains details in video sequence

data, Group of Picture (GOP) data, picture data, and macroblock data, to properly

decode the information in preparation for reconstruction and display of images);

means for generating a motion compensated reference value based on said

quantized data from said decoding means (col.6, ln.40-45 and fig.2, element 216 is a

motion compensator that generates a reference value based on the quantized data

from the VLD 212);  and,

means for adding the output from said IDCT means and said motion generating means to produce motion compensated pictures (col.6, ln.38-50; fig.2, element 215 is an adder that adds the output of the IDCT 214, S4, and the output of motion compensation means 216, S6, to produce motion compensated pictures S5).

Florencio does not specifically disclose wherein said extracted header information includes a macroblock-type, a motion vector magnitude, a motion vector count, non-zero discrete cosine transformer(DCT) coefficients, and a coded block pattern (CBP) number from said decoded block-based data packets.

However, this limitation is well known to one of ordinary skill in the art, but if one is not convinced, then one can peruse Wittig since Wittig teaches the extracted header information includes a macroblock-type, a motion vector magnitude, a motion vector count, non-zero discrete cosine transformer(DCT) coefficients, and a coded block pattern (CBP) number from said decoded block-based data packets (col.3, ln.9-14; Wittig discloses the header information contains macroblock type, motion type and a plurality of motion vectors which includes motion vector magnitude and count, DCT type which includes non-discrete cosine transformer coefficients, and a coded block pattern number).

Therefore, it would have been obvious to one of ordinary skill in the art to combine Wittig's teaching of utilizing the header information with Florencio's video decoding method and system for obtaining relevant header information so as to decode MPEG video data streams more efficiently and quickly, and at the same time, preserving high image quality (col.1, ln.44-50).

Regarding claim 2, Florencio discloses the decoder of claim 1, further comprising means for storing the output of said adding means (fig.2, element 217 is the frame memory that stores the output of adder 215).

Regarding claim 3, Florencio discloses the decoder of claim 1, wherein said extracting means being operative to execute said prediction operation according to said extracted header information of the plurality said macroblocks (col.6, ln.62 to col.7, ln.2-8; the variable length decoder (VLD) 212 in decoder 210 extracts the header information from the quantized data and performs a prediction operation, by using this header information that contains the details of data from the macroblocks, to properly decode the information in preparation for reconstruction and display of images).

Regarding claim 7, Florencio discloses the decoder of claim 1, wherein said prediction operation is implemented using a processing unit and software which controls the operation of said processing unit (col.7, ln.9-25; fig.2, element 230 is a controller that uses the header data to implement a prediction operation using a processing unit 234 and software for properly processing the decoding of image data).

Claims 4 and 5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Florencio (6,125,147) and Wittig (6,011,498) in view of Tan (6,408,096).

Regarding claim 4, Florencio discloses the extraction of the header information from the quantized data from the decoding means and for performing a prediction operation according to predetermined criteria (col.6, ln.62 to col.7, ln.2-8; the variable length decoder (VLD) 212 in decoder 210 extracts the header information from the quantized data and performs a prediction operation, by using this header information

which contains details in video sequence data, Group of Picture (GOP) data, picture

data, and macroblock data, to properly decode the information in preparation for

reconstruction and display of images). Florencio and Wittig do not specifically disclose

wherein said prediction operation defines the computation load of said IDCT means

and said motion generating means.

However, Tan teaches that the prediction operation defines the computation load

of the IDCT means and the motion generating means (col.8, ln.23-60; in fig.1, element

180 has the complexity estimator is applied to the decoder 190 and a feedback loop is

used to recursively estimate the complexity or to *selectively adjust* or *predict* the

complexity, ie. the computational load of the IDCT and motion generating means, of

the next decoded picture, wherein fig.6, the specifics of the complexity estimator are

further disclosed in that the error 186 of the complexity measure is used by calculating

the difference between the estimated complexity signal 184 and the actual complexity

signal 185, and that the error 186 goes back to the feedback gain 187, and then the

output of the feedback gain eventually goes back to the complexity estimator 183 for

constant recalculation of the complexity or the computational load of the IDCT and the

motion generating means, and the complexity information is implemented into decoder

190 of fig.2 where the components of the decoder 190 are affected or *scaled*, so the

IDCT 194 is *scaled* and the motion generating means 196 is *scaled*, thus, the

computational load of the MPEG decoder is *selectively adjusted*).

Therefore, it would have been obvious to one of ordinary skill in the art to

combine Tan's complexity estimator with Florencio and Wittig for properly decoding as

much video data as possible without losing pertinent information, and for versatile

decoding in multiple formats and resolutions (Tan col.8, ln.63 to col.9, ln.4).

Regarding claim 5, Florencio discloses the extraction of the header information

from the quantized data from the decoding means and for performing a prediction

operation according to predetermined criteria (col.6, ln.62 to col.7, ln.2-8; the variable

length decoder (VLD) 212 in decoder 210 extracts the header information from the

quantized data and performs a prediction operation, by using this header information

which contains details in video sequence data, Group of Picture (GOP) data, picture

data, and macroblock data, to properly decode the information in preparation for

reconstruction and display of images). Florencio and Wittig do not specifically disclose

wherein the computation load of said MPEG decoder is selectively adjusted by scaling

said IDCT means and said motion generating means based on said prediction

operation.

However, Tan teaches the computation load of the MPEG decoder is selectively

adjusted by scaling the IDCT means and the motion generating means based on the

prediction operation (col.1, ln.29-30, MPEG compression/decompression is used; col.8,

ln.23-60 and fig.1, element 180, the complexity estimator is applied to the decoder 190

and a feedback loop is used to further estimate the complexity or to *selectively adjust*

or *predict* the complexity, ie. the computational load of the IDCT and motion generating

means, of the next decoded picture, wherein fig.6, the specifics of the complexity

estimator are further disclosed in that the error 186 of the complexity measure is used

by calculating the difference between the estimated complexity signal 184 and the

actual complexity signal 185, and that the error 186 goes back to the feedback gain

187, and then the output of the feedback gain eventually goes back to the complexity

estimator 183 for constant recalculation of the complexity or the computational load of

the IDCT and the motion generating means, and the complexity information is

implemented into decoder 190 of fig.2 where the components of the decoder 190 are

affected or *scaled*, so the IDCT 194 is *scaled* and the motion generating means 196 is

*scaled*, thus, the computational load of the MPEG decoder is *selectively adjusted*).

Therefore, it would have been obvious to one of ordinary skill in the art to

combine Tan's complexity estimator with Florencio and Wittig for properly decoding as

much video data as possible without losing pertinent information, and for versatile

decoding in multiple formats and resolutions (Tan col.8, ln.63 to col.9, ln.4).


3.      Claims 8-11, 13-15 and 22-23 are rejected under 35 U.S.C. 103(a) as being

unpatentable over Florencio (6,125,147) in view of Tan (6,408,096).

Regarding claim 8, Florencio discloses a programmable video decoding system

(col.7, ln.9-25; fig.2, element 230 is a controller that uses the header data to implement

a prediction operation using a processing unit 234 and software for properly processing

or programming the decoding of image data), comprising:

a variable length decoder (VLD) configured to receive and decode a stream of

block-based data packets, said VLD being operative to output quantized data from said

decoded data packets (col.6, ln.25-27 and fig.2, element 212 is a variable length

decoder, VLD, that decodes compressed data and outputs quantized data);

a variable length decoder configured to extract the header information from said

block-based data packets (col.6, ln.62 to col.7, ln.2-8; the variable length decoder

(VLD) 212 in decoder 210 extracts the header information from the quantized data and

performs a prediction operation, by using this header information which contains details

in video sequence data, Group of Picture (GOP) data, picture data, and macroblock

data, to properly decode the information in preparation for reconstruction and display of

images);

an inverse quantizer coupled to receive the output of said variable length

decoder to operatively inverse quantize the quantized data received from said variable

length decoder (col.6, ln.28-31 and fig.2, element 213);

an inverse discrete cosine transformer (IDCT) coupled to the output of said

inverse quantizer for transforming the dequantized data from frequency domain to

spatial domain (fig.2, element 214);

a motion compensator (MC) configured to receive motion vector data from said

quantized data and to generate a reference signal (col.6, ln.40-45 and fig.2, element

216 is a motion compensator that generates a reference signal S6 based on the

quantized data from the VLD 212); and,

an adder for receiving said reference signal and said spatial domain data from

said IDCT to form motion compensated pictures (col.6, ln.38-50; fig.2, element 215 is

an adder that adds the output of the IDCT 214, S4, and the output of motion

compensation means 216, S6, to produce motion compensated pictures S5).

Florencio does not specifically disclose a complexity estimator configured to

extract the header information from said block-based data packets and further

configured to execute a video complexity algorithms based on said extracted header

information. However, Tan teaches a complexity estimator configured to extract the

header information from the block-based data packets and further configured to

execute a video complexity algorithms based on the extracted header information

(col.8, ln.23-32; in fig.1, at the encoding side, the complexity parameters are inserted

into the header information section of the block-based data packets, and then, at the

decoding side, the complexity parameter information in the header is sent to the

complexity estimator 180 in order to extract the header information with the complexity

parameter data in preparation for affecting, scaling or *configuring* the decoder 190 to

properly decode the video images by executing video complexity algorithms based on

the extracted header information, also in col.1, ln.18-21 and col.8, ln.13-22, Tan

teaches the implementation of software decoding or video complexity algorithms for

decoding).

Therefore, it would have been obvious to one of ordinary skill in the art to

combine Tan's complexity estimator with Florencio's video decoder method and

system for properly decoding as much video data as possible without losing pertinent

information, and for versatile decoding in multiple formats and resolutions (col.8, ln.63

to col.9, ln.4).

Florencio does not specifically disclose an estimator for calculating a total

computation load for said IDCT and said MC based on the classification of said header

information. However, Tan teaches a complexity estimator for calculating a total computation load for the IDCT and the MC based on the classification of said header information (col.8, ln.23-60 and fig.1, element 180, the complexity estimator is applied to the decoder 190 and a feedback loop is used to further estimate the complexity or to *calculate* the complexity, ie. the total computation load of the IDCT and motion generating means, of the next decoded picture, wherein fig.6, the specifics of the complexity estimator are further disclosed in that the error 186 of the complexity measure is used by calculating the difference between the estimated complexity signal 184 and the actual complexity signal 185, and that the error 186 goes back to the feedback gain 187, and then the output of the feedback gain eventually goes back to the complexity estimator 183 for constant recalculation of the complexity or the computational load of the IDCT and the motion generating means, and the complexity information is implemented into decoder 190 of fig.2 where the components of the decoder 190 are affected or *scaled*, so the IDCT 194 is *scaled* and the motion generating means 196 is *scaled*, thus, the computational load of the decoding method and system is *calculated*; and also in col.8, ln.23-32, fig.1, at the encoding side, the complexity parameters are inserted into the header information section of the block-based data packets, and then, at the decoding side, the complexity parameter information in the header is sent to the complexity estimator 180 in order to extract the header information with the complexity parameter data so that *calculation* of the complexity or *total computation load* is done in preparation for affecting, scaling or

*configuring* the decoder 190 to properly decode the video images according to the

extracted header information of the decoded macroblock-based data packets).

Therefore, it would have been obvious to one of ordinary skill in the art to

combine Tan's complexity estimator with Florencio's video decoder method and

system for properly decoding as much video data as possible without losing pertinent

information, and for versatile decoding in multiple formats and resolutions (col.8, ln.63

to col.9, ln.4).


Regarding claim 9, Florencio discloses a buffer configured to store the output of

the adder (fig.2, element 217 is the frame memory that stores the output of adder 215).

Regarding claim 10, Florencio discloses the variable length decoder configured

to extract the header information from the block-based data packets (col.6, ln.62 to

col.7, ln.2-8; the variable length decoder (VLD) 212 in decoder 210 extracts the header

information from the quantized data in the block-based data packets originating from

the video stream data). Florencio does not specifically disclose wherein said

complexity estimator being operative to execute said video complexity algorithms

according to the header information of said decoded block-based data packets, and

wherein the computation load of said decoding system is regulated by scaling said

IDCT and said MC based on said video complexity algorithms.

However, Tan teaches the complexity estimator being operative to execute the

video complexity algorithms according to the header information of the decoded block-

based data packets (col.8, ln.23-32; in fig.1, at the encoding side, the complexity

parameters are inserted into the header information section of the block-based data

packets, and then, at the decoding side, the complexity parameter information in the

header is sent to the complexity estimator 180 in order to extract the header information

with the complexity parameter data in preparation for affecting, scaling or *configuring*

the decoder 190 to properly decode the video images by executing video complexity

algorithms according to the extracted header information of the decoded block-based

data packets, also in col.1, ln.18-21 and col.8, ln.13-22, Tan teaches the implementation

of software decoding or video complexity algorithms for decoding), and

wherein the computation load of the decoding system is regulated by scaling the

IDCT and the MC based on the video complexity algorithms (col.8, ln.23-60 and fig.1,

element 180, the complexity estimator is applied to the decoder 190 and a feedback

loop is used to further estimate the complexity or to *regulate* the complexity, ie. the

computational load of the IDCT and motion generating means, of the next decoded

picture, wherein fig.6, the specifics of the complexity estimator are further disclosed in

that the error 186 of the complexity measure is used by calculating the difference

between the estimated complexity signal 184 and the actual complexity signal 185, and

that the error 186 goes back to the feedback gain 187, and then the output of the

feedback gain eventually goes back to the complexity estimator 183 for constant

recalculation of the complexity or the computational load of the IDCT and the motion

generating means, and the complexity information is implemented into decoder 190 of

fig.2 where the components of the decoder 190 are affected or *scaled*, so the IDCT 194

is *scaled* and the motion generating means 196 is *scaled*, thus, the computational load

of the decoding system is *regulated*, where also in col.1, ln.18-21 and col.8, ln.13-22,

Tan teaches the implementation of software decoding or video complexity algorithms for

decoding).

Therefore, it would have been obvious to one of ordinary skill in the art to

combine Tan's complexity estimator with Florencio's video decoder method and system

for properly decoding as much video data as possible without losing pertinent

information, and for versatile decoding in multiple formats and resolutions (col.8, ln.63 to

col.9, ln.4).

Regarding claim 11, Florencio discloses the variable length decoder configured

to extract the header information from the block-based data packets (col.6, ln.62 to

col.7, ln.2-8; the variable length decoder (VLD) 212 in decoder 210 extracts the header

information from the quantized data in the block-based data packets originating from

the video stream data). Florencio does not specifically disclose the computation load

of said IDCT and said MC are selectively adjusted according to said video complexity

algorithms.

However, Tan teaches the computation load of said IDCT and said MC are

selectively adjusted according to said video complexity algorithms (col.8, ln.23-60 and

fig.1, element 180, the complexity estimator is applied to the decoder 190 and a

feedback loop is used to further estimate the complexity or to *selectively adjust* the

complexity, ie. the computational load of the IDCT and motion generating means, of

the next decoded picture, wherein fig.6, the specifics of the complexity estimator are

further disclosed in that the error 186 of the complexity measure is used by calculating

the difference between the estimated complexity signal 184 and the actual complexity

signal 185, and that the error 186 goes back to the feedback gain 187, and then the

output of the feedback gain eventually goes back to the complexity estimator 183 for

constant recalculation of the complexity or the computational load of the IDCT and the

motion generating means, and the complexity information is implemented into decoder

190 of fig.2 where the components of the decoder 190 are affected or *selectively*

*adjusted*, so the IDCT 194 is *scaled or selectively adjusted* and the motion generating

means 196 is *scaled or selectively adjusted*, thus, the computational load of the

decoding system is *selectively adjusted* based on the video , where also in col.1, ln.18-

21 and col.8, ln.13-22, Tan teaches the implementation of software decoding or video

complexity algorithms for decoding).

Therefore, it would have been obvious to one of ordinary skill in the art to

combine Tan's complexity estimator with Florencio's video decoder method and

system for properly decoding as much video data as possible without losing pertinent

information, and for versatile decoding in multiple formats and resolutions (col.8, ln.63

to col.9, ln.4).

Regarding claim 13, Florencio discloses the system where decoding is

implemented using a processing unit and software, which controls the operation of the

processing unit (col.7, ln.9-25; fig.2, element 230 is a controller that uses the header

data to implement a prediction operation using a processing unit 234 and software for

properly processing the decoding of image data). Florencio does not specifically

disclose the video complexity algorithm is implemented using a processing unit and

software, which controls the operation of the processing unit. However, Tan teaches

the video complexity algorithm is implemented using a processing unit and software,

which controls the operation of the processing unit (in col.1, ln.18-21 and col.8, ln.13-

22, Tan teaches the implementation of software decoding or video complexity

algorithms for decoding, and that since Tan's decoding operation clearly occurs on a

computer with the use of software and hardware, Tan must disclose a processing unit

in the computer hardware otherwise the software would not run without the hardware

components of the computer).

Therefore, it would have been obvious to one of ordinary skill in the art to

combine Tan's complexity estimator with Florencio's video decoder method and system

for properly decoding as much video data as possible without losing pertinent

information, and for versatile decoding in multiple formats and resolutions (col.8, ln.63 to

col.9, ln.4).

Regarding claim 14, Florencio discloses a method for improving decoding

efficiency of an encoded digital video signal employing an MPEG digital video decoder

(see fig.2 and col.2, ln.28-29) having a variable length code decoder (fig.2, element

212), an inverse quantizer (fig.2, element 213), an inverse discrete cosine transformer

(fig.2, element 214), and a motion compensator (fig.2, element 216), said method

comprising the steps of:

receiving a compressed video data stream at said VLC decoder and producing

decoded data therefrom (col.6, ln.25-27 and fig.2, element 212 is a variable length

decoder, VLD, that receives the compressed video data stream, decodes compressed

data and outputs quantized data);

retrieving header information from said decoded data (col.6, ln.62 to col.7, ln.2-8;

the variable length decoder (VLD) 212 in decoder 210 extracts the header information

from the quantized data and performs a prediction operation, by using this header

information which contains details in video sequence data, Group of Picture (GOP)

data, picture data, and macroblock data, to properly decode the information in

preparation for reconstruction and display of images);

dequantizing said decoded data using said inverse quantizer(IQ) to generate

dequantized, decoded data (col.6, ln.28-31 and fig.2, element 213);

employing said IDCT for transforming said dequantized, decoded data from

frequency domain to spatial domain to produce difference data (fig.2, element 214

produces difference data S4);

employing said MC for generating a reference data of said encoded digital video

signal (col.6, ln.40-45 and fig.2, element 216 is a motion compensator that generates a

reference data S6 based on the quantized data from the VLD 212); and,

combining said reference data and said difference data to produce motion

compensated pictures (col.6, ln.38-50; fig.2, element 215 is an adder that adds or

combines the difference data S4 and the reference data S6 to produce motion

compensated pictures S5).

Florencio does not specifically disclose an estimator for calculating a total

computation load for said IDCT and said MC based on the classification of said header

information. However, Tan teaches a complexity estimator for calculating a total

computation load for the IDCT and the MC based on the classification of said header

information (col.8, ln.23-60 and fig.1, element 180, the complexity estimator is applied

to the decoder 190 and a feedback loop is used to further estimate the complexity or to

*calculate* the complexity, ie. the total computation load of the IDCT and motion

generating means, of the next decoded picture, wherein fig.6, the specifics of the

complexity estimator are further disclosed in that the error 186 of the complexity

measure is used by calculating the difference between the estimated complexity signal

184 and the actual complexity signal 185, and that the error 186 goes back to the

feedback gain 187, and then the output of the feedback gain eventually goes back to

the complexity estimator 183 for constant recalculation of the complexity or the

computational load of the IDCT and the motion generating means, and the complexity

information is implemented into decoder 190 of fig.2 where the components of the

decoder 190 are affected or *scaled*, so the IDCT 194 is *scaled* and the motion

generating means 196 is *scaled*, thus, the computational load of the decoding method

and system is *calculated*; and also in col.8, ln.23-32, fig.1, at the encoding side, the

complexity parameters are inserted into the header information section of the block-

based data packets, and then, at the decoding side, the complexity parameter

information in the header is sent to the complexity estimator 180 in order to extract the

header information with the complexity parameter data so that *calculation* of the

complexity or *total computation load* is done in preparation for affecting, scaling or

*configuring* the decoder 190 to properly decode the video images according to the

extracted header information of the decoded macroblock-based data packets).

Therefore, it would have been obvious to one of ordinary skill in the art to

combine Tan's complexity estimator with Florencio's video decoder method and

system for properly decoding as much video data as possible without losing pertinent

information, and for versatile decoding in multiple formats and resolutions (col.8, ln.63

to col.9, ln.4).

Regarding claim 15, Florencio discloses retrieving header information from the

decoded data (col.6, ln.62 to col.7, ln.2-8; the variable length decoder (VLD) 212 in

decoder 210 extracts the header information from the quantized data and performs a

prediction operation, by using this header information which contains details in video

sequence data, Group of Picture (GOP) data, picture data, and macroblock data, to

properly decode the information in preparation for reconstruction and display of

images). Florencio does not specifically disclose comprising the step of transmitting

said total computation load to said IDCT and said MC.

However, Tan teaches the step of transmitting the total computation load to the

IDCT and the MC (col.8, ln.23-60 and fig.1, element 180, the complexity estimator is

applied to the decoder 190 and a feedback loop is used to further estimate or calculate

the complexity, ie. the total computational load of the IDCT and motion generating

means, of the next decoded picture, wherein fig.6, the specifics of the complexity

estimator are further disclosed in that the error 186 of the complexity measure is used

by calculating the difference between the estimated complexity signal 184 and the

actual complexity signal 185, and that the error 186 goes back to the feedback gain

187, and then the output of the feedback gain eventually goes back to the complexity

estimator 183 for constant recalculation of the complexity or the computational load of

the IDCT and the motion generating means, and the complexity information or the *total*

*computation load* is implemented into decoder 190 of fig.2 where the components of

the decoder 190 are affected or *scaled*, so the IDCT 194 is *scaled* and the motion

generating means 196 is *scaled*, thus, the total computational load is *transmitted* to the

IDCT and the MC).

Therefore, it would have been obvious to one of ordinary skill in the art to

combine Tan's complexity estimator with Florencio's video decoder method and

system for properly decoding as much video data as possible without losing pertinent

information, and for versatile decoding in multiple formats and resolutions (col.8, ln.63

to col.9, ln.4).

Regarding claim 22, Florencio discloses a prediction method of improving

decoding efficiency of an encoded data video signal employing an MPEG digital video

decoding system (fig.2 and col.2, ln.28-29) having a variable length code decoder

(fig.2, element 212), an inverse quantizer (fig.2, element 213), an inverse discrete

cosine transformer (fig.2, element 214), a motion compensator (fig.2, element 216),

and an estimator, said method comprising the steps of:

decoding a compressed bitstream including a plurality of macroblocks to obtain a

corresponding decoded macro block (col.6, ln.25-27 and fig.2, element 212 is a

variable length decoder, VLD, that receives the compressed video data stream,

decodes compressed data and outputs quantized data and decoded macroblock); and

obtaining a header classification criteria from the header information of said

decoded macroblock (col.6, ln.62 to col.7, ln.2-8; the variable length decoder (VLD)

212 in decoder 210 extracts the header information from the quantized data and

performs a prediction operation, by using this header information which contains details

or *header classification criteria* in video sequence data, Group of Picture (GOP) data,

picture data, and macroblock data, to properly decode the information in preparation

for reconstruction and display of images).

Florencio does not specifically disclose an estimator employing said estimator for

predicting a total computation load according to the header information from said

decoded macroblock and forwarding said total computation load to said IDCT and said

MC; and, adjusting the computation load of said IDCT and said MC according to said

total computation load.

However, Tan teaches an estimator employing the estimator (fig.1, element 180

where in fig.6, the specifics of element 180 is shown in that element 183 is a

complexity estimator) for predicting a total computation load according to the header

information from the decoded macroblock (col.8, ln.23-32; in fig.1, at the encoding side,

the complexity parameters are inserted into the header information section of the

macroblock based data packets, and then, at the decoding side, the complexity

parameter information in the header is sent to the complexity estimator 180 in order to

extract the header information with the complexity parameter data so that *calculation* of

the complexity or *total computation load* is done in preparation for affecting, scaling or

*configuring* the decoder 190 to properly decode the video images) and

forwarding the total computation load to the IDCT and the MC (col.8, In.23-60

and fig.1, element 180, the complexity estimator is applied to the decoder 190 and a

feedback loop is used to further estimate or calculate the complexity, ie. the total

computational load of the IDCT and motion generating means, of the next decoded

picture, wherein fig.6, the specifics of the complexity estimator are further disclosed in

that the error 186 of the complexity measure is used by calculating the difference

between the estimated complexity signal 184 and the actual complexity signal 185, and

that the error 186 goes back to the feedback gain 187, and then the output of the

feedback gain eventually goes back to the complexity estimator 183 for constant

recalculation of the complexity or the computational load of the IDCT and the motion

generating means, and the complexity information or the *total computation load* is

implemented into decoder 190 of fig.2 where the components of the decoder 190 are

affected or *scaled*, so the IDCT 194 is *scaled* and the motion generating means 196 is

*scaled*, thus, the total computational load is *forwarded* to the IDCT and the MC); and,

adjusting the computation load of the IDCT and the MC according to the total

computation load (col.8, In.23-60 and fig.1, element 180, the complexity estimator is

applied to the decoder 190 and a feedback loop is used to further estimate or calculate

the complexity, ie. the total computational load of the IDCT and motion generating

means, of the next decoded picture, wherein fig.6, the specifics of the complexity

estimator are further disclosed in that the error 186 of the complexity measure is used

by calculating the difference between the estimated complexity signal 184 and the

actual complexity signal 185, and that the error 186 goes back to the feedback gain

187, and then the output of the feedback gain eventually goes back to the complexity

estimator 183 for constant recalculation or re-*adjustment* of the complexity or *the*

*computational load of the IDCT and the motion generating means (MC)*, and the

complexity information or the *total computation load* is implemented into decoder 190

of fig.2 where the components cf the decoder 190 are affected or *scaled*, so the IDCT

194 is *scaled* and the motion generating means 196 is *scaled*, thus, the computation

load of the IDCT and MC is *adjusted* accordingly to the total computational load).

Therefore, it would have been obvious to one of ordinary skill in the art to

combine Tan's complexity estimator with Florencio's video decoder method and

system for properly decoding as much video data as possible without losing pertinent

information, and for versatile decoding in multiple formats and resolutions (col.8, ln.63

to col.9, ln.4).

Regarding claim 23, Florencio teaches the decoding of the decoded macroblock

according to the header information (col.6, ln.62 to col.7, ln.8; note VLD 212 decodes

the decoded macroblock according to the header information and sends the fidelity

indicative parameter "DATA" where "DATA" contains "other information" regarding the

video frames) together with motion vector information received in the compressed

bitstream (col.6, ln.25-27, in fig.2, VLD 212 has the motion vector information received

in the compressed bitstream, as evidenced by the exit of MV or motion vectors from

element 212). Florencio does not specifically disclose further comprising the step of

decoding the decoded macroblock according to the total computation load, together

with motion vector information received in the compressed bitstream.

However, Tan teaches the step of decoding the decoded macroblock according

to the total computation load, together with motion vector information received in the

compressed bitstream (col.8, ln.13-22 and in fig.1, note the decoder 190, the

macroblock data is decoded according to the complexity or the total computational load

as estimated by complexity estimator 180, and that the decoder 190 has a VLD 191, in

fig.2, that receives the motion vector information from the compressed bitstream stored

at the buffer 170, and so the decoder 190 uses the total computation load, calculated

by complexity estimator 180, together with the motion vector information received in

the compressed bitstream).

Therefore, it would have been obvious to one of ordinary skill in the art to

combine Tan's complexity estimator with Florencio's video decoder method and

system for properly decoding as much video data as possible without losing pertinent

information, and for versatile decoding in multiple formats and resolutions (col.8, ln.63

to col.9, ln.4).

Claims 12 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable

over Florencio (6,125,147) and Tan (6,408,096) as applied to claims 8 and 22 above,

and further in view of Wittig (6,011,498).

Regarding claims 12 and 24, Florencio discloses the extraction of the macroblock

header information (col.6, ln.62 to col.7, ln.2-8; the variable length decoder (VLD) 212 in

decoder 210 extracts the header information from the quantized data and performs a

prediction operation, by using this header information which contains details of the macroblock data, to properly decode the information in preparation for reconstruction and display of images). Also, Tan discloses the extraction of header information (col.8, ln.23-32). Florencio and Tan do not specifically disclose wherein said extracted header information includes a macroblock-type, a motion vector magnitude, a motion vector count, non-zero discrete cosine transformer(DCT) coefficients, and a coded block pattern (CBP) number from said decoded block-based data packets.

However, this limitation is well known to one of ordinary skill in the art, but if one is not convinced, then one can peruse Wittig since Wittig teaches the extracted header information includes a macroblock-type, a motion vector magnitude, a motion vector count, non-zero discrete cosine transformer(DCT) coefficients, and a coded block pattern (CBP) number from said decoded block-based data packets (col.3, ln.9-14; Wittig discloses the header information contains macroblock type, motion type and a plurality of motion vectors which includes motion vector magnitude and count, DCT type which includes non-discrete cosine transformer coefficients, and a coded block pattern number).

Therefore, it would have been obvious to one of ordinary skill in the art to combine Wittig's teaching of utilizing the header information with Florencio and Tan's combined video decoding method and system for obtaining relevant header information so as to decode MPEG video data streams more efficiently and quickly, and at the same time, preserving high image quality (col.1, ln.44-50).

## *Allowable Subject Matter*

4.      Claims 16-21 and 25-30 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

5.      The following is a statement of reasons for the indication of allowable subject matter:  Florencio discloses a method and apparatus for reducing breathing artifacts in compressed video.  Tan discloses a method for computational graceful degradation in an audiovisual compression system.  Wittig discloses a generic dual-speed variable length decoding architecture for MPEG-2 video data.  Neither Florencio, Tan, Wittig, nor any of the prior art does not disclose details or specifics of the combination of limitations as disclosed in claims 16 and 25: wherein the step of calculating and predicting said total computation load further comprising the steps of: determining a first computation load based on the macroblock of said retrieved header information; determining a second computation load based on the motion magnitude of said retrieved header information; determining a third computation load based on the motion vector magnitude of said retrieved header information; determining a fourth computation load based on the number of non-zero DCT coefficients of said retrieved header information; determining a fifth computation load based on the coded block pattern number of said retrieved header information; and, combining said first, second, third, fourth, fifth computation loads, and an average computation load to obtain said total computation load.  Dependent claims 17-21 and 26-30 contain allowable subject matter for the same reasons as claims 16 and 25.

## *Conclusion*

6.      **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action.  In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action.  In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the mailing date of this final action.

### *Contact Information*

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Allen Wong whose telephone number is (703) 306-5978.

The examiner can normally be reached on Mondays to Thursdays from 8am-6pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Christopher Kelley can be reached on (703) 305-4856.  The fax phone

number for the organization where this application or proceeding is assigned is 703-

872-9306.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free).

Allen   Wong
Examiner
Art Unit 2613

AW
8/12/04

CHRIS KELLEY
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2000